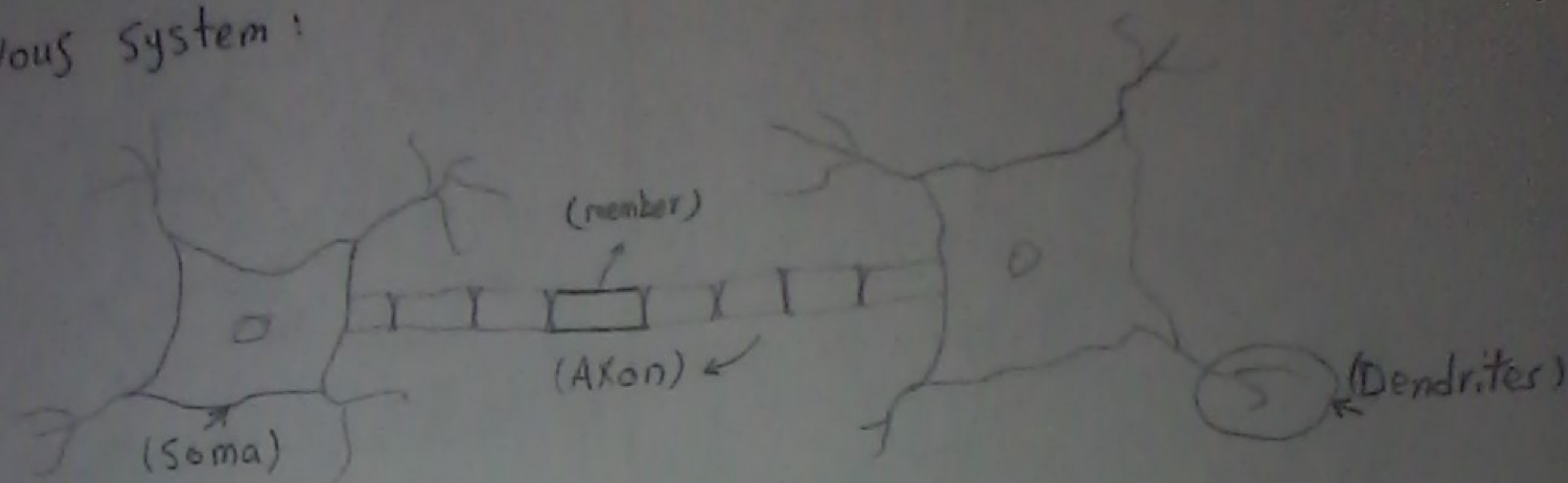


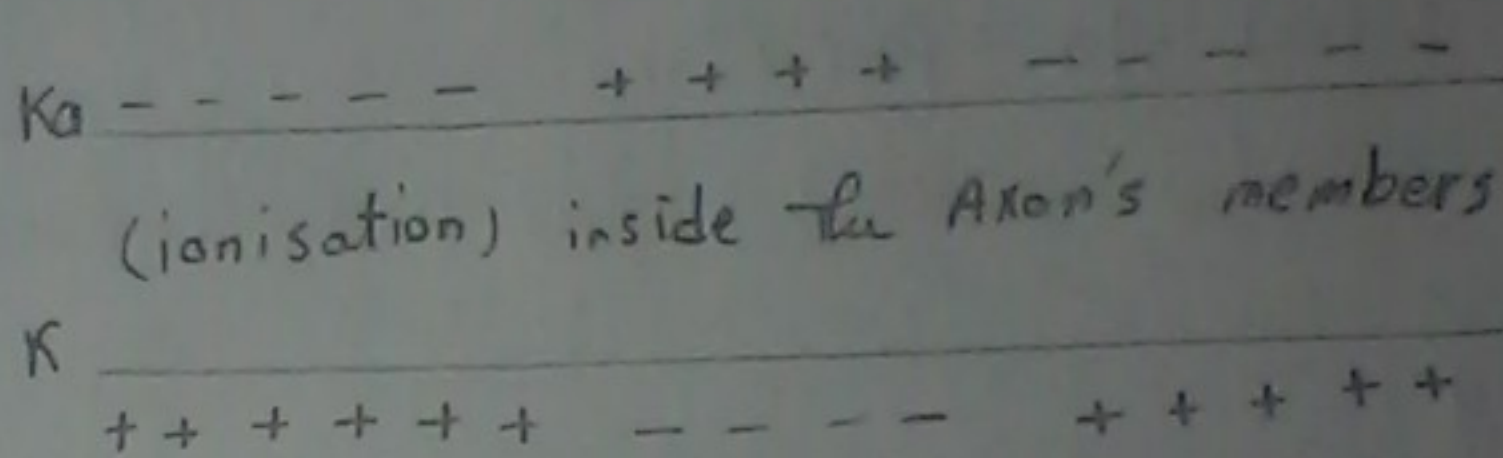
Chapter (I) : From biological to ANN (Artificial Neural Network)

* Nervous System :



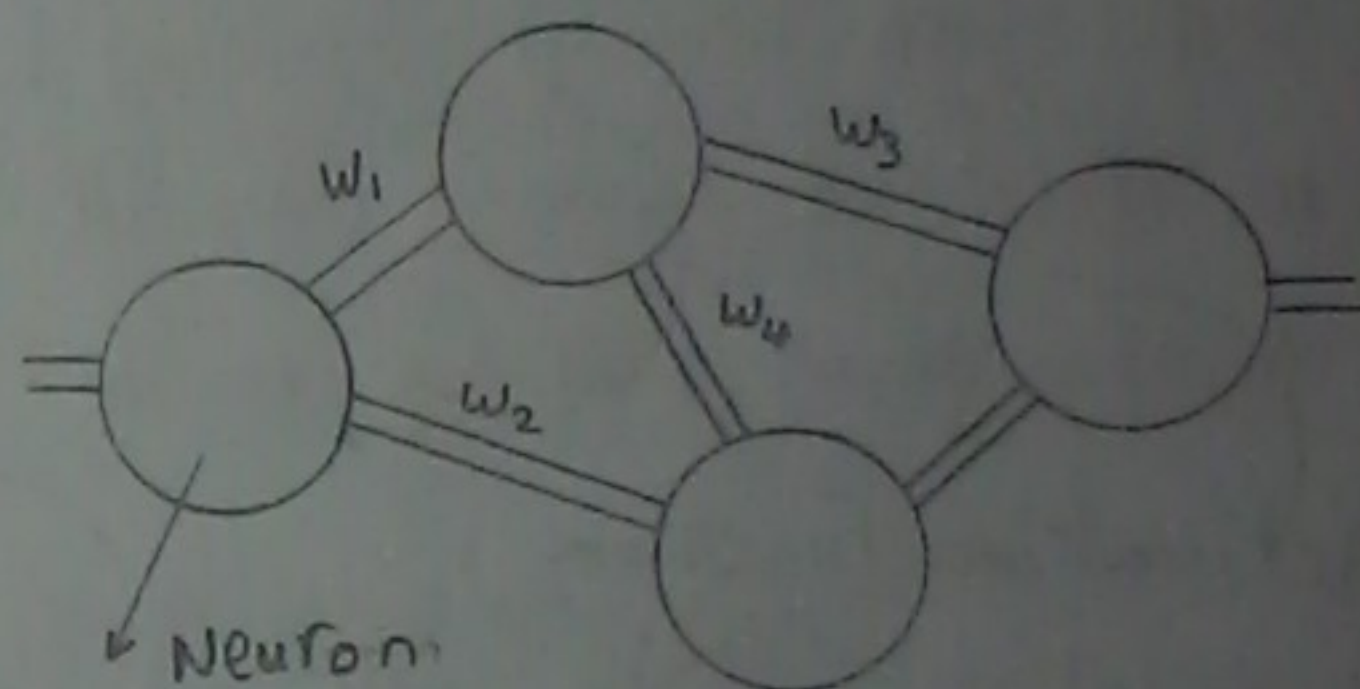
- Soma : (generate fixed signals)
- Axon : (Carrying signals (ions))

* Chemical reactions (ion-Pump) occurs in Axon



* Neural Network :

- Neuron \rightarrow Soma
- Layers \rightarrow Axon
- Threshold \rightarrow chemical reaction
- weight \rightarrow chemical reaction



ANN : (Art. Neural Network)

- I/P layer
- Hidden layer (Neurons)
- O/P layer

* Active function :

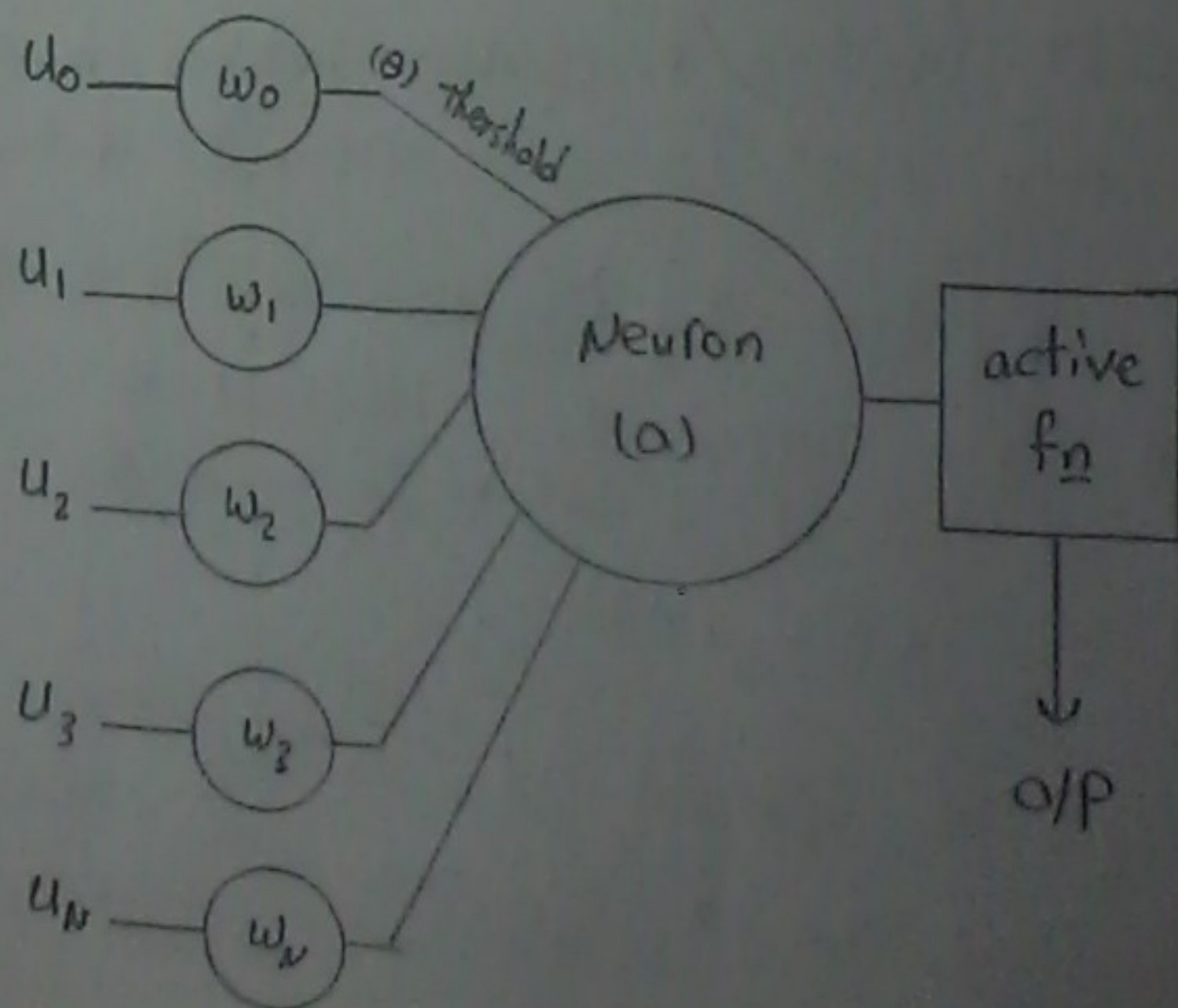
$$a = \sum_{n=1}^{n=N} u_n w_n + \text{Threshold}$$

$$= \sum_{n=0}^{n=N} u_n w_n \quad (\text{with threshold inside})$$

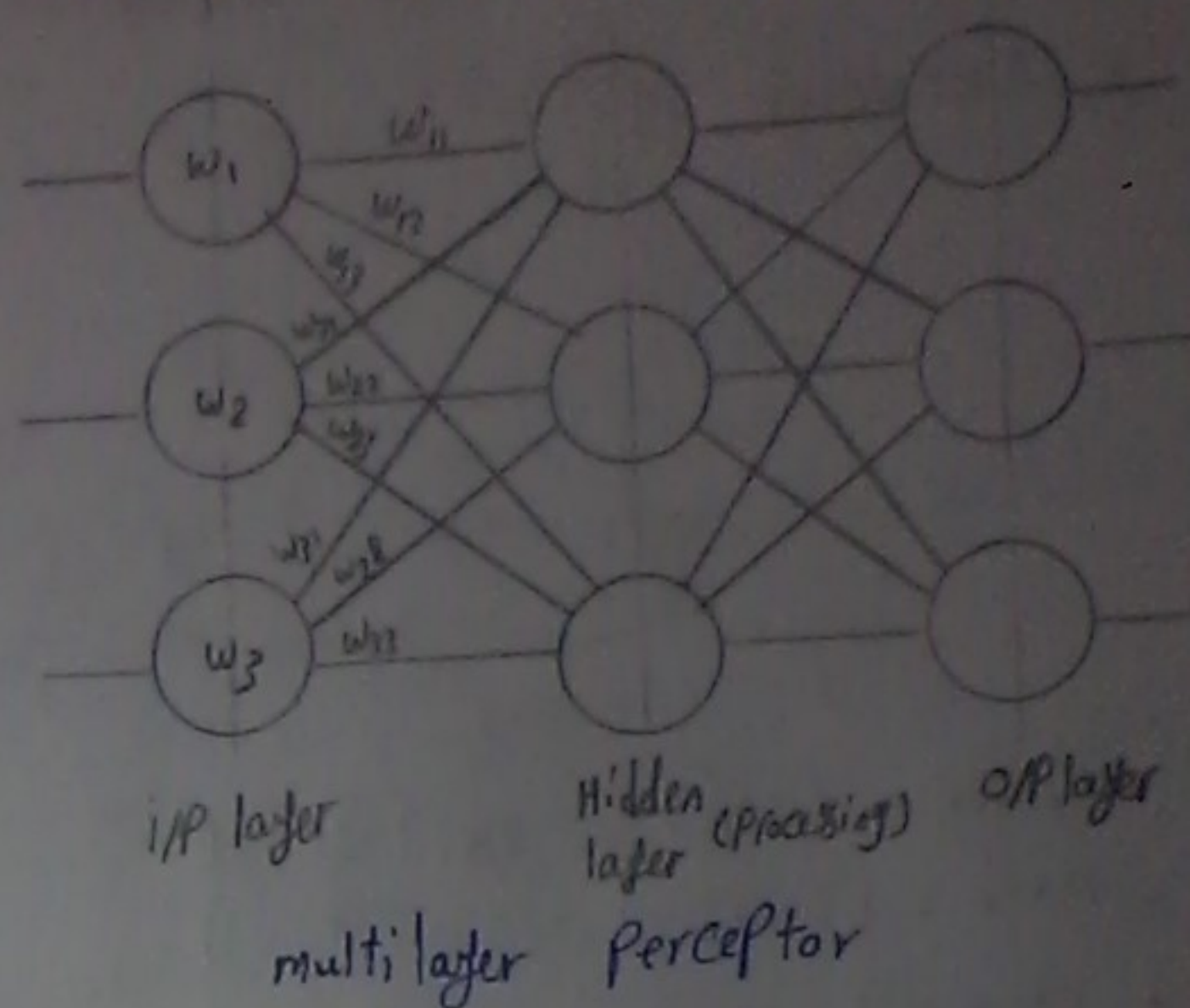
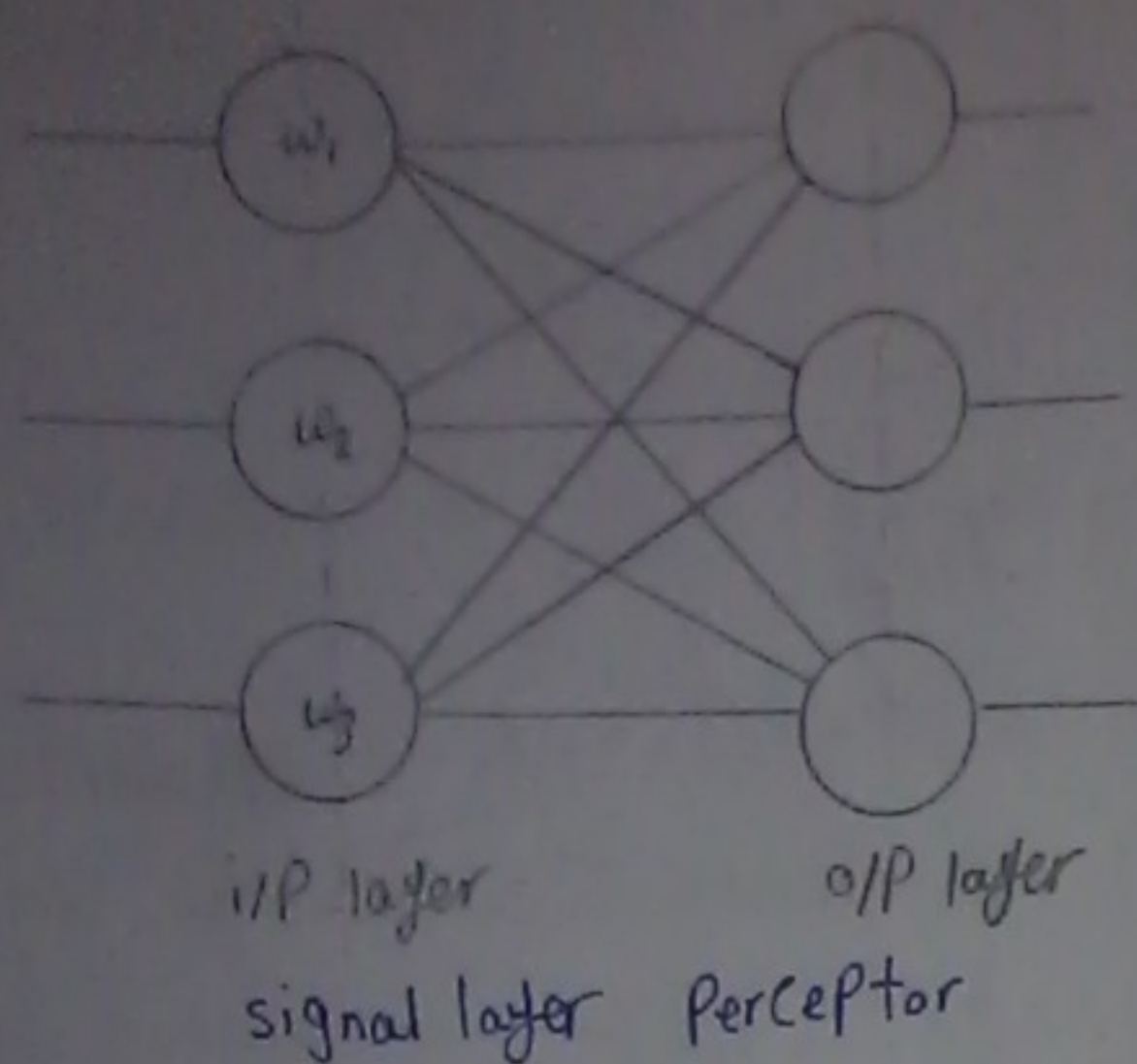
$$\theta = u_0 w_0$$

* (in matrix) :

$$a = U^T W + \theta$$



Architecture of NN:



* w is adapted according threshold (activation function)

Learning:

- Supervised Learning (training for NN)
- unsupervised Learning (No i/P - o/P relation)
- Reinforcement learning (No i/P: The i/P generated inside NN)

Activation Function:

- A relation between o/P and (i/Ps & weights)

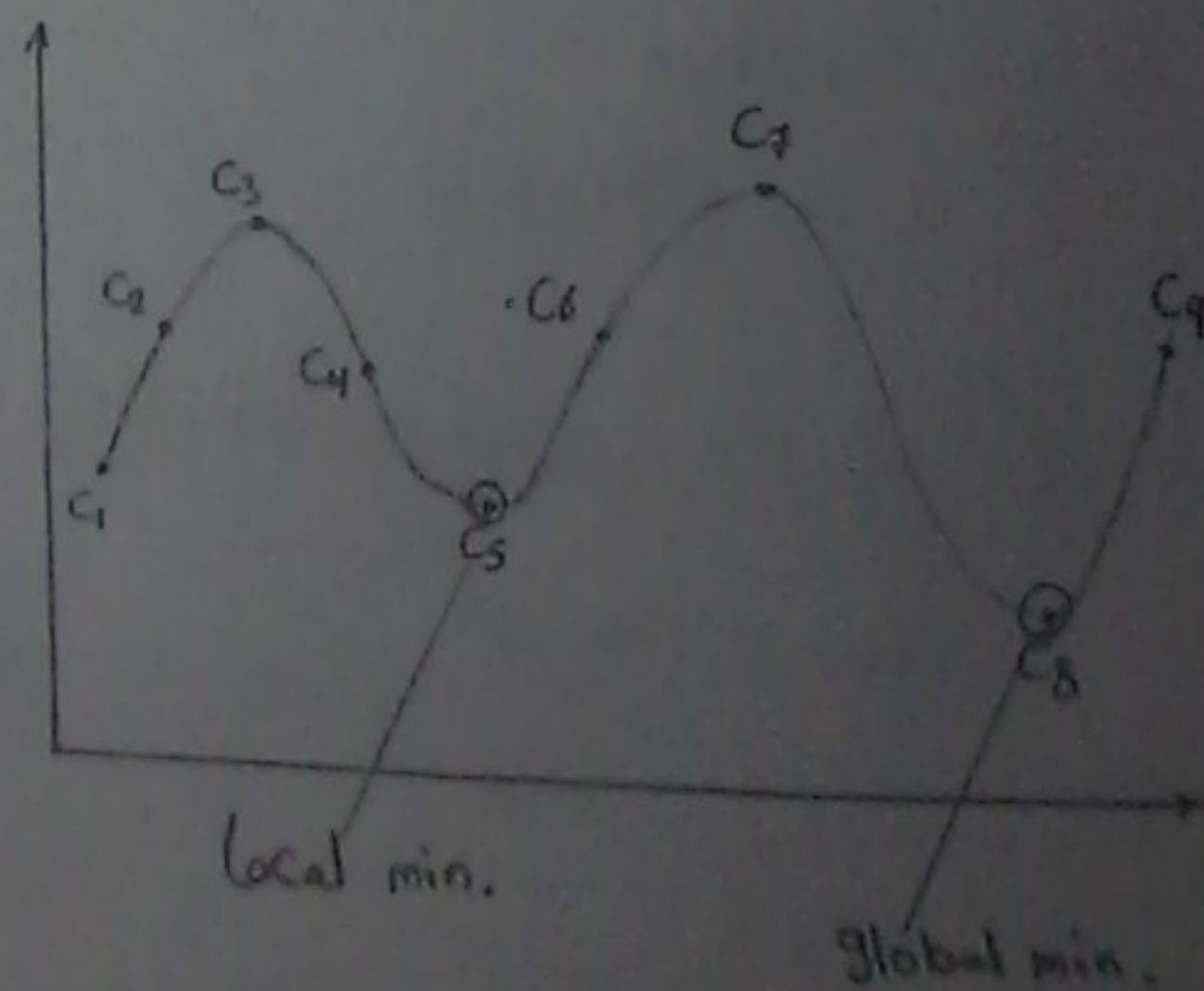
* Cost function: Compare between measured output & actual output
- min. cost fn \rightarrow optimal solution.

$$C = E[f(x) - y] \text{ (The mean value of the difference between measured o/p \& actual o/p)}$$

* NN maybe trapped at C_5 : Think it the best sol. but if it continue to C_8 is the best sol. (NN Problem)

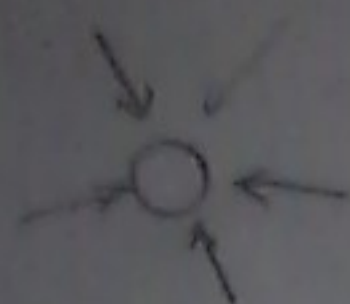
* to get the optimal sol. we take

$$C \text{ gradient to } w \quad \frac{\partial C}{\partial w} \approx 0$$



* For Ideal Solution : (Dynamic states) (Phase space)

- Convergent



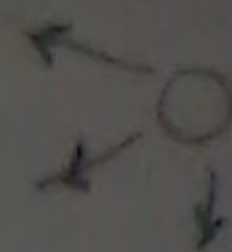
O/P Converge to ideal state

- cyclic (oscillating)



oscillate and return to same point

- chaotic



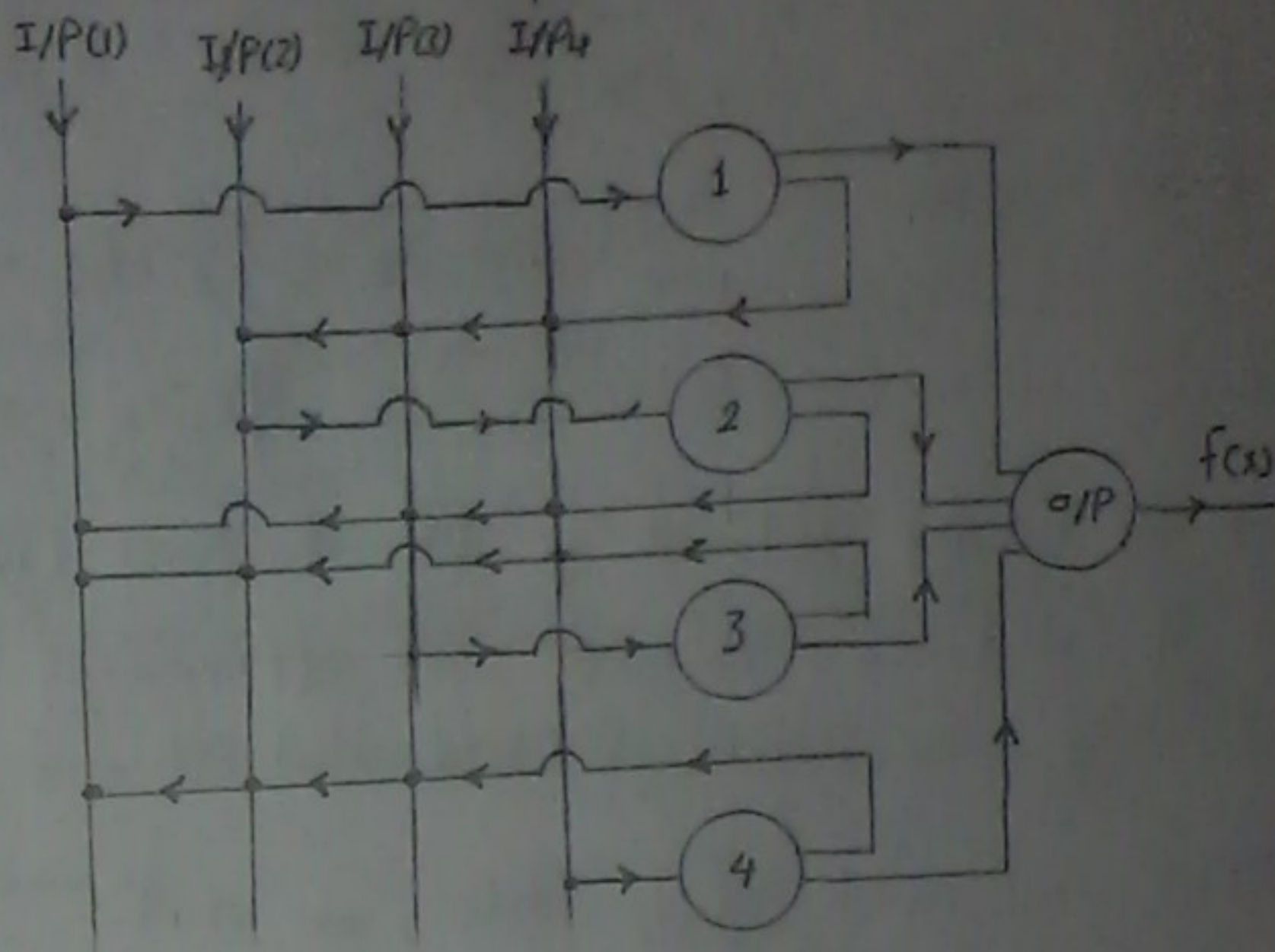
get far from ideal solution

Recurrent NN : (Chapter II)

- O/P of each N backwards to all components except itself.

- $w_{ii} = 0$

- $w_{ij} = w_{ji}$ (symmetry)



Applications of NN :

- Classifications (Pattern Recognition) (Image signal processing)

- Data Processing

- Control

- Robotics

- Medical applications

- Speech Recognition

- Picture Recognition

note: The main Parameter is the weight of each interconnection, based on application & activation function.

- output is function in activation function $\rightarrow x(t) = f(a)$

- activation f_n
 Linear
 Threshold or
 Ramp

the target from RNN

- unique solution
- Existing solution
- stable state

$$\left. \begin{array}{l} \text{unique solution} \\ \text{Existing solution} \end{array} \right\} x(t) \rightarrow \frac{dx(t)}{dt} = F(x(t))$$

$$= F(x_1(t), x_2(t), \dots, x_n(t))$$

* unique solution \rightarrow restriction

(certain condition to reach st. state for each Neuron not for whole Network (w adaptive))

$$x_1(t_1) \rightarrow w_1$$

$$x_1(t_2) \rightarrow w_2$$

\vdots

$$x_1(t_n) \rightarrow w_n$$

$$\Rightarrow \text{distance} = |x_1(t_2) - x_1(t_1)|$$

$$= |x_1(t_3) - x_1(t_2)|$$

* operating network in different times (iteration)

* if distance decrease, we approaching unique solution

$d \rightarrow \min_{(0)} \rightarrow \text{toward ideal sol.} \rightarrow \text{steady state}$

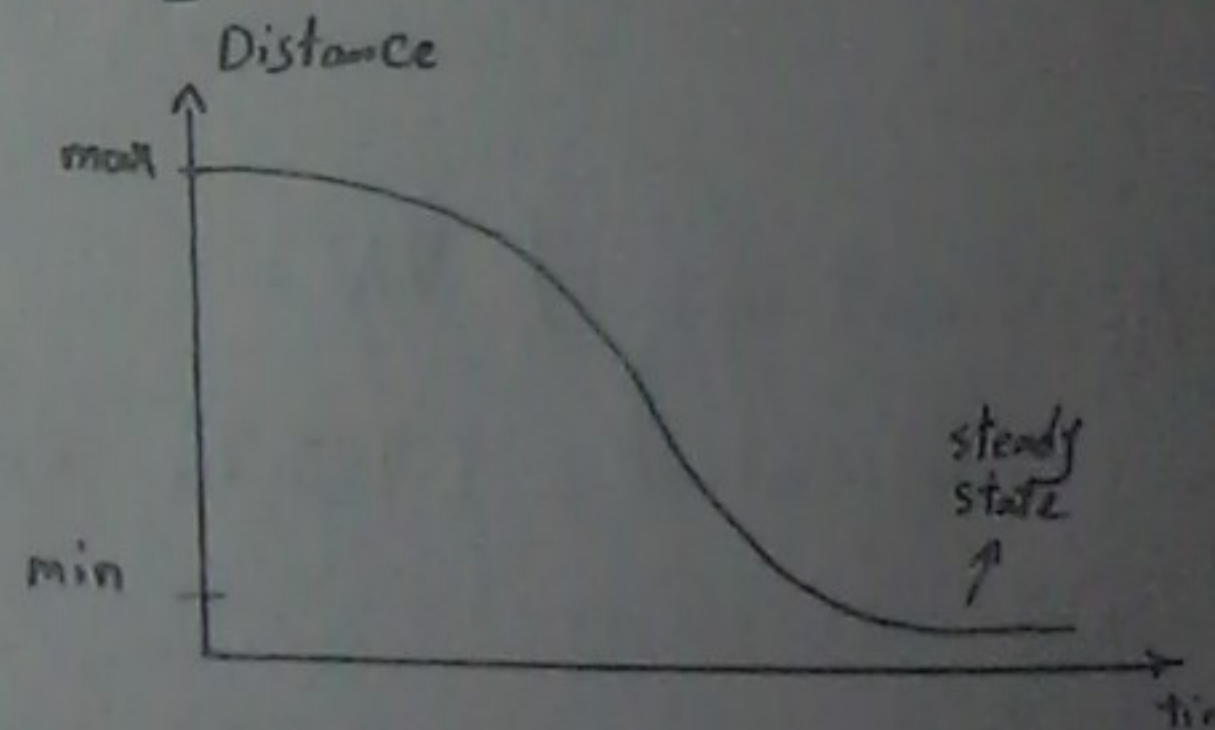
- After we reach steady state, we start testing & validation.

- If we get S.S far from zero, so if we change time weights will not change

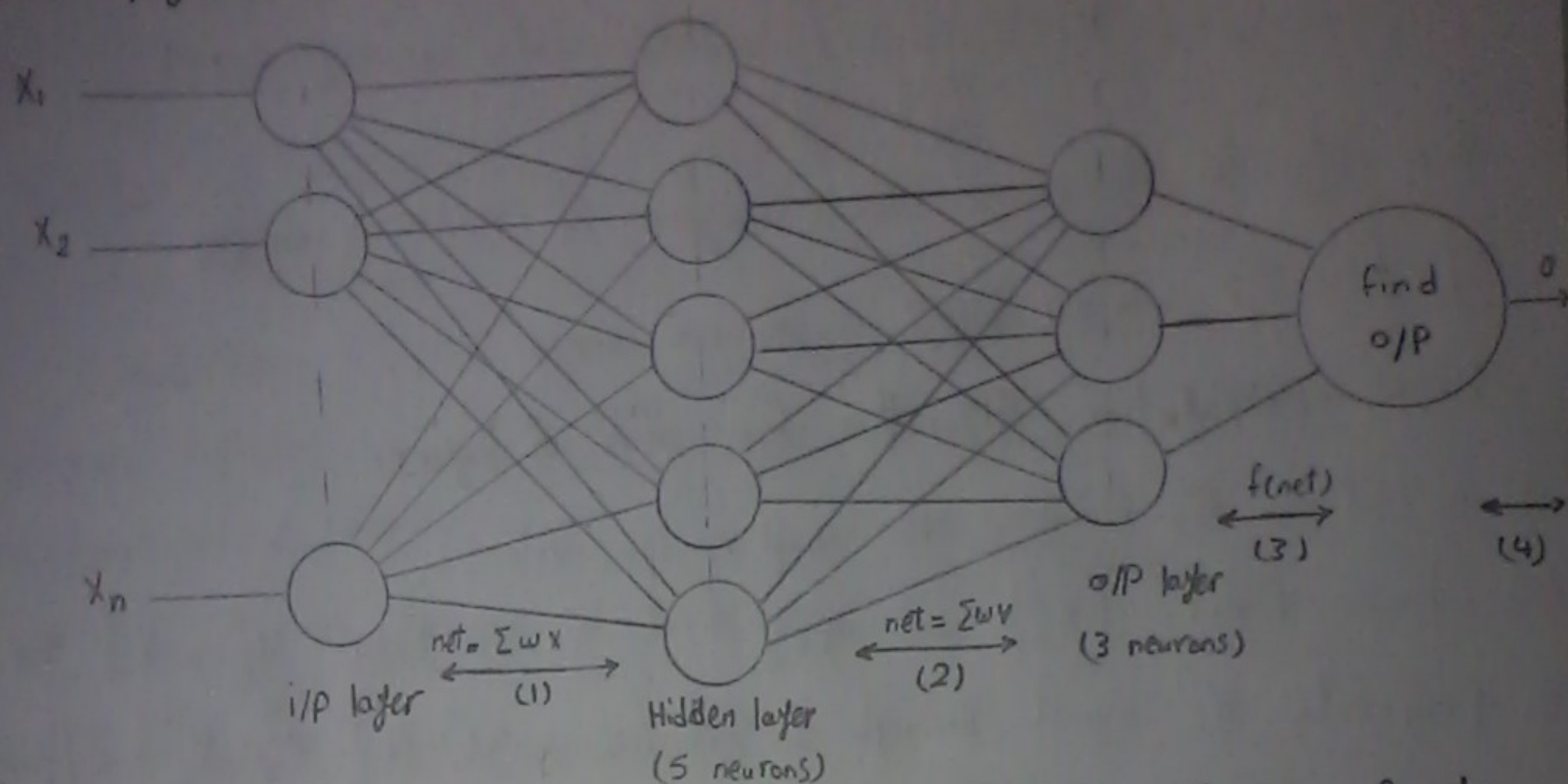
- Return and reConstruct Network.

note: Large no. of neurons cause more processing, more time to get S.S.

(NN Problem)



Back Propagation Neural Network (BPNN)



- the relation between i/p \Rightarrow layers \Rightarrow o/p comes from the activation function:

* i/p x_d , o/p o_d for neuron j , it will receive the following function:

$$\text{net}_j^d = \sum_{k=1}^5 w_{jk} x_k^d \quad \text{--- (1)}$$

$$V_j^d = f(\text{net}_j^d) = f\left(\sum_{k=1}^5 w_{jk} x_k^d\right) \quad \text{--- (2)}$$

* o/p (at neuron i)

$$\text{net}_i^d = \sum_{j=1}^3 w_{ij} V_j^d \quad \text{--- (3)}$$

$$= \sum_{j=1}^3 w_{ij} \cdot f\left(\sum_{k=1}^5 w_{jk} x_k^d\right) \quad \text{--- (3)}$$

* final o/p (o): $o_i^d = f(\text{net}_i^d) = f\left(\sum_{j=1}^3 w_{ij} \cdot f\left(\sum_{k=1}^5 w_{jk} x_k^d\right)\right) \quad \text{--- (4)}$

i/p

note:

* In Traditional NNs, the o/p changes to reach ideal case only by try & Error.
 - Nowadays, there are another techniques like optimization, genetic algorithm, ...) To obtain the ideal (optimal) weights for this network to reach or be close to ideal solution

$$E[w] = \frac{1}{2} \sum_{d=1}^4 \sum_{i=1}^2 (t_i^d - o_j^d)^2$$

desired o/p
actual o/p

then $\frac{\partial E}{\partial w}$ to determine the weights of minimum energy

$E = f(o - T) \rightarrow$ our target is minimum energy

- Back-propagation Algorithm for m # of layers:

(1) initialize the weights to small random values.

(2) Choose pattern (X_K^d) and apply it to the i/p layer $V_K^0 = X_K^d$ for all K

(3) Propagate the signal through the network $\Rightarrow V_i^m = f(\sum_j w_{ij}^m V_j^{m-1}) = f(\text{net}_i^m)$
 \downarrow
preceding layer o/p

(4) Compute delta for the output layer

$$\delta_i^m = f'(\text{net}_i^m) (t_i^d - V_i^d)$$

\downarrow error
 \downarrow derivative
 \downarrow Target
 \downarrow actual
 \Rightarrow related to cost fn (Energy fn)

(5) Compute δ at preceding layers

$$\delta_i^{m-1} = f'(\text{net}_i^{m-1}) (\sum_j w_{ij}^m \delta_j^m)$$

(6) update the connections (weights)

$$\Delta w_{ij}^m = \eta \delta_i^m V_j^{m-1} \Rightarrow \delta w_i \rightarrow \text{learning rate}$$

$$(w_{ij}^{\text{new}} = w_{ij}^{\text{old}} + \Delta w_{ij}) \Rightarrow \text{new weight}$$

Radial Base function Network (RBFN)

- Special case of BPNN when activation function is radial Basis

* Advantages of RBFN .

- No Local minimum
- No oscillation
- Trained rapidly

* Disadvantages of RBFN :

- After training it becomes slow.

Basis function : $\phi = |x_i - x_j|$ distance (euclidian distance)

• i/p it self

$$\phi(x) = \sum_{i=1}^N w_i (|x_i - c_i|)$$

$N \rightarrow N \text{ space}$ centre

→ The activation function in radial Basis .

Types of Radial .

- Gaussian
- Multi-Quadric
- Inverse-Quadric
- Inverse multi-Quadric

} must be symmetric
(distance between point and other point must be the same)

Pattern Recognition (Face Recognition using NN)

* Face Recognition: A science based on image processing

- image \rightarrow image processing \rightarrow features

* main procedures:

- Capture the image
- Image enhancement
- Features Extraction
- Features selection (significant feature)

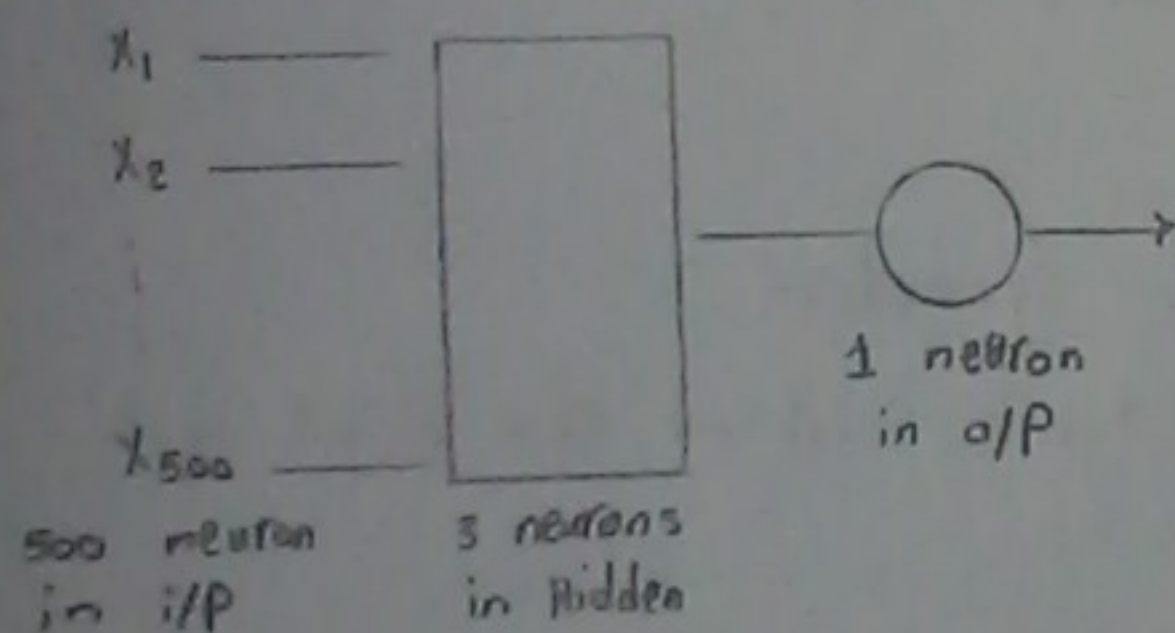
features {
- color
- shape
- age
- length
;

Example: Picture with 500 pixels.

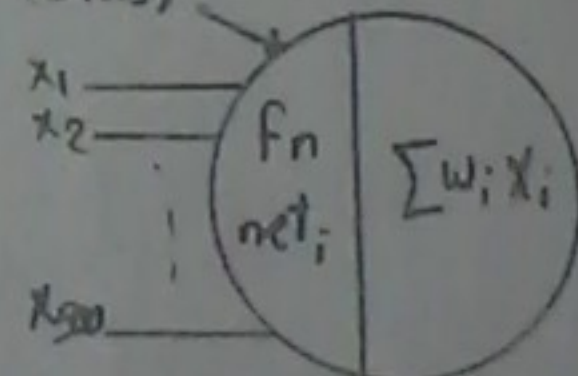
- the main feature is color intensity of pixel

- 500 i/p neurons (no. of pixels) , 1 o/p neurons (recognition u, d, R, L)

0 \leftarrow \rightarrow 255
Black white



(1) Training (learning by an algorithm (BPNN))



(2) Validation (test from data base)

(3) Test (test out of data base)

- * up \rightarrow down (false)
- up \rightarrow up (true)
- up \rightarrow Right (False)

\rightarrow For some situations (no. of false = no. of True) we can calculate Confusion matrix & Then the accuracy.

Examples:

- Classification between a Car and a Van
- Classification between happy, sad and neutral face
- Classification direction of a face

- What is Neural Network mean? (main Idea / Concept)

Neural Networks (NNs) also known as Artificial Neural Networks (ANN), are massively parallel interconnection network of simple (usually adaptive) elements and their hierarchical organization which are intended to interact with objects of real world in the same way as biological nervous systems do.